Course Project

Financial Analysis on Twitter

Part 1. Exploratory Data Analysis

You will be provided with a dataset including over 5 million tweets in a certain time period. The tweets mention at least one of the stocks traded in the use markets as well as primitive information about the users making those tweets. You will also be given financial information about those stocks. The task is to first perform a thorough explanatory data analysis to gain a brief and holistic vision on the data you have at hand and what you can do with it.

Bonus points are awarded for creative visualizations.

Your results should include explanatory plots plus additional notes on the analysis of the following items. These are the deliverables of the first part.

- Statistics on most and least tweeted stocks. Perform segmentation of the companies based on the number of tweets they have. Provide relevant visualizations.
- Statistics on distributions of 5 individual stocks over time. Choose the individual stocks to perform reflect different sectors of the economy.
- Statistics on distributions of all financial tweets over time.
- Statistics on distributions of retweets per tweets including individual stocks (at least 2 chosen stocks) over time.
- Statistics on most important financial information on individual stocks (at least 2 chosen stocks) computed solely from the financial information (not tweets).

- Time series movement directions through time for individual stocks (at least 2). Choose companies you are familiar with. Try to explain the reason behind these directions from real world news.
- Co-occurrence of various stocks in the same tweets.

Part 2. Sentiment Analysis

In this part, you are supposed to perform sentiment analysis on a dataset of labeled tweets. You are given a dataset of labeled tweets, on which you should perform the following steps (For each step, define a pipeline, so that the cleaning steps become reusable):

- Clean the data
  - Removing duplicate values and useless data (both columns and rows)
  - Handling upper/lower case, etc.
- Construct a pipeline with techniques tough in the course (Stemming, lemmatization, etc.)
- Perform feature extraction with both bag of words and TF-IDF
- Train a supervised algorithm on each of the above features. This step includes:
  - Splitting train/validation/test set
  - Testing different models and evaluate the performance on validation set.
  - Choosing 2-3 candidates for hyper parameter tuning and performing the relevant steps.
  - Finalizing the model choice and reporting the final result on test set. Beware you can only choose the test set once.
- Use a pre-trained library like BERT or SpaCy for sentiment analysis.
- Use the API of ChatGPT and construct a classifier with either prompt engineering or function calling. Using complex structures in this part such as initial classification with in-house model and validation with ChatGPT API has bonus points. **To access the API of ChatGPT if you haven't got one,** write a document and calculate the exact usage rate you will have from it in train, validation, and test phase (this document is necessary as we have to make adjustments to

provide you with API key). Send the document to @amirsoleix and in case of approval, you will be provided with an API key to use. Limitations and constraints will be given to you. **Don't postpone this to days near deadline (meaning send your document at least 5 days before the time you want to use it) as this requires arrangement and tracking. In case of happening, there is no guarantee on delivery of the API key and no points will be given to this part.**

- **Instead of using ChatGPT, you can choose to fine-tune one of the models on HuggingFace. This will come with a huge bonus point if done correctly!**
- Compare the four methods (Bag of Words, TF-IDF, Pre-trained, ChatGPT API or HuggingFace) methods and discuss the results.

Some helpful libraries are TextBlob, NLTK, SpaCy, OpenAI.

Part 3. Sentiment Analysis

In the first two parts, you were given a dataset of financial tweets and trained a few classifiers to extract sentiment from those tweets.

In this part you have to complete the following steps:

- Apply two best classifiers of the second part on the dataset of the first part.
- Using different correlation measures (at least 2 and you can use libraries), find the correlation between the sentiments of each CashTag and its value (from the financial data in the companies dataset) in a time interval.
- Using at least two different correlation measures, find the correlation between change of sentiments of a CashTag and the number of tweets related to that CashTag in the time series (By time series of tweets, we mean the number of tweets per given time that was calculated in the first part).
- Find examples where the classifiers do not agree on the sentiment of the tweet and analyze the results and discuss where does each classifier make mistakes. This part is mostly for the first 3 classifiers, because as we all know, **when ChatGPT makes classification mistakes, it is really hard to solve!** Try to provide methods on how to make classifications better. **Implementation of your ideas and reaching a better result yields bonus point based on the amount of effort and complexity of your methodology.**

- Discuss transfer learning and fine-tuning and how they can be used to improve the overall effect of the project. Use your limited results from the second part.

If you do not have access to appropriate hardware, you can use Google Colab for implementation. If that is also inconvenient, try to select a part of the dataset that is reflective of the entire population and perform the steps on that part.

Bonus Time

1. Read the article **“Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter”.** Try to replicate the methodology on the article and reach the results mentioned.

In case of success, the bonus point is 35% of the project grade. Applicable to quizzes and other homework.

2. Choose a website from the following options. Research on which python libraries you can use for scraping the web and write a python script that scraps the items in the website and creates a CSV file with appropriate columns that collects the data on the website.

- https://www.digikala.com/search/notebook-netbook-ultrabook/

- https://www.zillow.com/homes/for_rent/apartment_duplex_type/

- https://divar.ir/s/tehran

- https://zapier.com/apps

The following capabilities of the script yields the following bonus points:

- **Plain simple extraction of data 5%**

- **If you script performs scrolling the page to load more content or clicks on buttons to load the next page 5%.**

- **If you script clicks on each of the items and scraps the results page thoroughly 5%.**

This is not as hard as it seems. **Give it a shot!**